

Appendix A

```
import java.awt.*;
5  import java.awt.event.*;
import javax.swing.*;
import java.io.*;

public class CSApplet extends JApplet
10 {
    private String image;
    private double physicalWidth;
    private double physicalHeight;
    public void init()
15 {
        super.init();

        physicalWidth = 0.0;
        physicalHeight = 0.0;
20
        // Read in the values of the parameters from
        the html page
        // The parameters are image, physicalWidth, and
        physicalHeight
25
        image = getParameter("image");

        String param = getParameter("physicalWidth");
30
        if (param != null)
        {
            physicalWidth = Double.parseDouble(param);
        }

        param = getParameter("physicalHeight");
35
        if (param != null)
        {
            physicalHeight =
40 Double.parseDouble(param);
        }

        // Add a panel to the applet where the image
        will be drawn
45
        Container contentPane = getContentPane();

        try
        {
            // Create the panel and pass it the html
50 parameter values
```

```

        contentPane.add(new CSPanel(image,
physicalWidth, physicalHeight));
    }
    catch (IOException e)
5      {
      }
    }
}

10  import java.awt.*;
    import java.awt.event.*;
    import javax.swing.*;
    import java.util.*;
    import java.io.*;

15  class CSPanel extends JPanel
    {
        private String imageName;
        private double physicalWidth;
        private double physicalHeight;
20      private int imageResX;
        private int imageResY;
        private Image image;
        CSPanel(String imageName, double physicalWidth,
25      double physicalHeight) throws IOException
        {
            this.imageName = imageName;
            this.physicalWidth = physicalWidth;
            this.physicalHeight = physicalHeight;

30            // Hack for reading the xResolution and
            yResolution from the JPEG header
            FileInputStream in = new FileInputStream(new
            File(imageName));
35            DataInputStream din = new DataInputStream(in);

            // Skip over the first seven shorts
            for (int i = 0; i < 7; i++)
            {
40                din.readShort();
            }

            // get the x/y resolution
            imageResX = din.readShort();
45            imageResY = din.readShort();

            // close streams
            din.close();
            in.close();

50            // read in the image

```

```

        image =
Toolkit.getDefaultToolkit().getImage(imageName);

        // wait until the image is entirely read in
5      MediaTracker tracker = new MediaTracker(this);
        tracker.addImage(image, 0);

        try
        {
10          tracker.waitForID(0);
        }
        catch (InterruptedException e)
        {
        }
15    }
    public void paintComponent(Graphics g)
    {
        super.paintComponent(g);

20      // get the logical screen resolution ie
        640x480, 1024x768
        Toolkit tk = Toolkit.getDefaultToolkit();
        Dimension screen = tk.getScreenSize();

25      // compute the physical size of the image
        double xInches = image.getWidth(this) /
        (double) imageResX;
        double yInches = image.getHeight(this) /
        (double) imageResY;
30      // compute the physical dpi of the display
        double xDPI = screen.width / physicalWidth;
        double yDPI = screen.height / physicalHeight;

35      // compute the logical width of the image
        double newWidth = xDPI * xInches;
        double newHeight = yDPI * yInches;

        // setup the graphics context
40      Graphics2D g2 = (Graphics2D) g;

        g2.setRenderingHint(RenderingHints.KEY_ANTIALIASING,
        RenderingHints.VALUE_ANTIALIAS_ON);

45      g2.setRenderingHint(RenderingHints.KEY_RENDERING,
        RenderingHints.VALUE_RENDER_QUALITY);

        // draw the image in the panel with scaling
        g2.drawImage(image, 0, 0, (int) newWidth, (int)
50      newHeight, null);
    }

```

```

}

<!doctype html public "-//w3c//dtd html 4.0 transitional//en">
<html>
5 <head>
    <meta name="GENERATOR" content="Mozilla/4.61 [en] (WinNT; I)
    [Netscape]">
    <title>TestApplet1</title>
</head>
10 <body>
    <object classid="clsid:8AD9C840-044E-11D1-B3E9-00805F499D93"
        width="100%" height="100%" align="middle"
        codebase="http://<object classid="clsid:8AD9C840-044E-11D1-B3E9-
00805F499D93"
15     width="100%"<param NAME="code" VALUE="CSApplet.class"><param
NAME="codebase" VALUE="classes/"><param NAME="type"
VALUE="application/x-java-applet;version=1.2.2"><param NAME="image"
value="d:/patent/moose.jpg"><param NAME="physicalWidth"
value="11.25"><param NAME="physicalHeight" value="8.5"><param
20 NAME="scriptable" VALUE="true">No
JDK 1.2 support for APPLET!!</object>
</body>
</html>
25

```